

August 01, 2018

# STRATEGIES TO DETERMINE WHETHER PERFORMANCE REQUIREMENTS CAN BE SATISFIED IN A 4G/5G/WLAN NETWORK

Mukesh Taneja

Vishal Murgai

Follow this and additional works at: [https://www.tdcommons.org/dpubs\\_series](https://www.tdcommons.org/dpubs_series)

---

## Recommended Citation

Taneja, Mukesh and Murgai, Vishal, "STRATEGIES TO DETERMINE WHETHER PERFORMANCE REQUIREMENTS CAN BE SATISFIED IN A 4G/5G/WLAN NETWORK", Technical Disclosure Commons, (August 01, 2018)  
[https://www.tdcommons.org/dpubs\\_series/1385](https://www.tdcommons.org/dpubs_series/1385)



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

## STRATEGIES TO DETERMINE WHETHER PERFORMANCE REQUIREMENTS CAN BE SATISFIED IN A 4G/5G/WLAN NETWORK

AUTHORS:

Mukesh Taneja  
Vishal Murgai

### ABSTRACT

Proposed herein are techniques to determine whether a 4G/5G base station, or a Wireless Local Area Network (WLAN) Access Point (AP), can meet/satisfy performance requirements of newly added (or handed over) applications or user equipment (UE) in a 4G/5G/WLAN system. The techniques presented herein collect a variety of parameters from different base stations, perform pre-processing, and analyze these parameters at a Performance Predictor Module. The Performance Predictor Module uses Machine Learning (ML) methods, such as Support Vector Machines (SVM) or Support Vector Regressions (SVR) with the feature set and target variables, to learn to predict whether or not a base station can meet performance requirements of, for example, added UE. The learning process is accelerated since the Performance Predictor Module learns from multiple base stations. The Performance Predictor Module also predicts degraded Quality of Service (QoS) classes that can be supported by the originating or neighboring base stations if the required/desired QoS class cannot be supported by the originating base station. This feedback is provided to the originating base station which takes policy based decisions using the feedback and the policies configured at that base station and the UE. The Performance Predictor Module may also provide feedback for various what-if scenarios.

### DETAILED DESCRIPTION

LTE/5G networks use Orthogonal frequency-division multiple access (OFDMA) and other related access technologies. For example, 802.11n/ac systems use OFDM and 802.11ax systems use OFDMA. Multi-user, multiple input, multiple output (MU-MIMO) is also used in some of these systems. As new user equipment (UE) joins a network (or an existing UE starts a new application), an LTE/5G base station needs to know whether or

not it can support the QoS required/desired by the UE. It is also useful to have this indication in a Wireless Local Area Network (WLAN) network when a station (STA) joins an access point (AP), starts a new application, is performing hand-over operations, *etc.* There are large number of varying parameters, such as parameters related to traffic models, channel models *etc.*, in these networks and it is difficult to determine whether or not performance requirements of a new application, UE, *etc.*, can be supported. Presented herein is an improved solution to these problems.

More specifically, proposed herein is a Performance Predictor Module that may run, for example, as part of a Self-Organizing Network (SON) server. For ease of description, the Performance Predictor Module is sometimes referred to herein as a “PPM-SON.” In operation, the PPM-SON collects various parameters from different 4G/5G base stations (or WLAN APs), performs data cleaning, and ingests these features into a Machine Learning (ML) model. This training set empowers the PPM-SON to learn whether or not a particular scenario results in meeting performance requirements (of applications / UEs) for a given base station and assists in admission control for a base station, as and when needed. PPM-SON also provides additional capabilities as detailed below. For ease of description, the terms base station (BS) and access point (AP) are used interchangeably herein. Similarly, also for ease of illustration, the terms user equipment (UE) and station (STA) are used interchangeably herein.

In certain examples, ML methods such as such as Support Vector Machines (SVM) or Support Vector Regressions (SVR) are used for learning purposes by the Performance Predictor module. Table 1, below, illustrates feature set for these ML methods in accordance with examples presented herein.

## Features for ML Methods for 4G/5G/WLAN systems

Parameters for each time interval ( $t_s, t$ ) where current time = $t$ and $t_s = t - \text{delta}$
Type of access technology supported by base station (such as LTE, 5G, 802.11ac/ax/..., ..)
Location coordinate of base station (or AP) and UEs (or STAs) in that cell
Base station vendor id and software (or algorithm) release id (as different software releases may use different algorithms)
Type (or QoS class) of apps running in that network for each UE (e.g. a UE could be running a delay sensitive app, such as video streaming, and some non-delay-sensitive apps)
Number of apps for each QoS class (for DL as well as UL) (For example, we could count number of apps for each QCI for an LTE network)
Max velocity of each UE (moving or stationary, speed if moving, ..) in a given time period, if available
Weighted average channel conditions of each UE (available in 4G/5G systems. May or may not be available in 802.11 systems)
Max transmit power of a base station
Max transmit power of each UE
Backhaul capacity of each base station
HW/SW Loading of a base station
% of resource blocks (such as normalized RUs in 802.11ax, PRBs in LTE, ..) being used for delay sensitive apps

**Table 1**

Target variable for the ML methods may include: (1) percentage of applications for each QoS class for which QoS requirements are (or can be) met by a given base station (or an AP); (2) QoS parameters observed for each application (as applicable), such as Latency, Throughput, Packet loss, jitter, *etc.*; (3) QoS class (or parameters) that can be supported by neighboring base stations, among others.

Tables 2 and 3, below, illustrate features, how those features are collected, and whether those features may be collected using new protocol extensions or existing protocols.

Feature	How is it collected?	New (protocol extensions) or Existing
Location coordinate of base station and UEs in cell	Provided by UE to BS (or BS computes using network based techniques). BS relays it to PPM-SON	Existing
QoS Classes/apps for each UE	Provided by UE to BS or detected at BS via Deep packet inspection or signaling mechanisms (as applicable) BS relays it to PPM-SON	Existing + New
Max transmit power of a base station and for each UE	Provided by UE to BS. BS relays it to PPM-SON	Existing + New
Velocity of each UE (moving or stationary, speed if moving (optional))	Provided by UE to BS. BS relays it to PPM-SON	Existing
Backhaul capacity for each base station	BS relays to PPM-SON	Existing
HW/SW loading for each base station (or some other load indicator)		Existing
Product or vendor id and software release id		Existing + New

**Table 2**

Feature	How its collected?	New (protocol extensions/telemetry etc) or Existing
Weighted average channel conditions - Mandatory for 4G/5G case - Optional for WLAN case	Provided by UE to BS per 3GPP spec. BS computes weighted average Relays to PPM-SON	Existing (over UE-BS) + maybe new over BS – PPM-SON server
% of (normalized) resource blocks (such as RUs in 802.11ax, PRBs in LTE systems, ...) used for delay sensitive apps	Measured by BS Relayed to PPM-SON	New
UL QoS: UL Throughput, UL Latency	Measured at BS by counting UL bytes over a period of time. We will add support for UE to provide incurred delay in UE queues to BS. This is done by adding vendor specific objects in the MAC (or some other protocols) running between UE and BS	New + Existing
DL QoS related parameters: Throughput, latency/jitter, loss	Measured at base station  Packet is timestamped on entry to a base station and this timestamp is updated until packet is served from that BS	New + Existing
Type of access technology supported by base station	BS relays to PPM-SON	Existing

**Table 3**

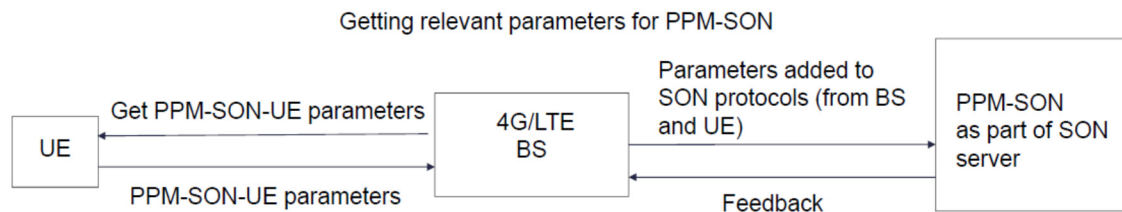
The various parameters may be measured (e.g., automatically) at the base station, UE, *etc.* and communicated to PPM-SON module. For example, as a new UE (or a new application for an existing UE) tries to join a 4G/5G/WLAN network, the 4G/5G base station (or WLAN AP) contacts the PPM-SON module, provides the required parameters, and is informed (by the PPM-SON) as to whether or not the base station would be able to support that application for required/desired performance. This base station uses this feedback from the PPM-SON module, along with its own policies, to automatically decide whether or not to admit that new UE (or new application of an existing UE), thereby providing a closed loop.

DL QoS related parameters (such as throughput, latency/jitter, and loss) are measured at the base station. Packets are timestamped on entry to a base station and this timestamp is updated until packet is served from that BS. For UL QoS, UL throughput is measured at the BS by counting UL bytes over a period of time. For UL latency/jitter, the techniques support UE to provide incurred delay in UE queues to the BS. This is done by adding vendor specific objects in the MAC (or some other protocols) running between UE and BS. With weighted average channel conditions in a 4G/5G case, the UE keeps informing base station about its channel conditions as per 3GPP specs. The BS computes weighted average

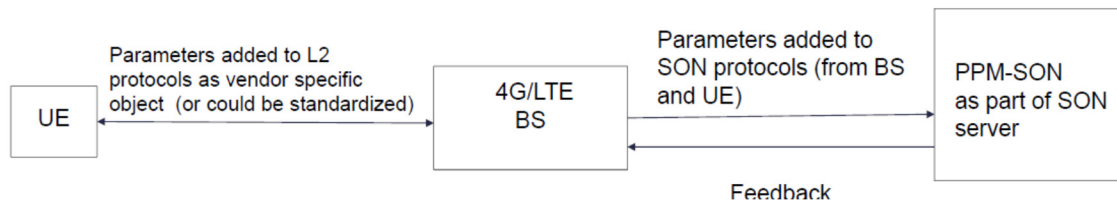
channel conditions for each UE and informs to PPM-SON module. Support of this for WLAN case is optional in the techniques presented herein.

Regarding the percentage of (normalized) resource blocks (such as RUs in 802.11ax, PRBs in LTE systems,...) used for delay sensitive applications, these are measured and provided by the base station to PPM-SON module. The velocity of UE may optionally be provided by the UE to the BS which, in turn, provides it to the PPM-SON. The location of the UE is also provided by the UE to BS which again provides to the PPM-SON. Alternatively, network based location techniques can be used by BS to measure location of UE and provide to PPM-SON. Configured parameters of BS (such as software id, location id, max Tx power) are provided by BS to the PPM-SON. In general, the BS uses some protocols for OAM purposes to communicate with the SON server. Objects can be added in those protocols to communicate these parameters.

FIG. 1, below, is a schematic diagram generally illustrating how PPM-SON obtains relevant parameters.



PPM-SON-UE parameters: Need to get only those parameters that are not available with existing protocols (such as UL latency incurred in UL queues at a UE)

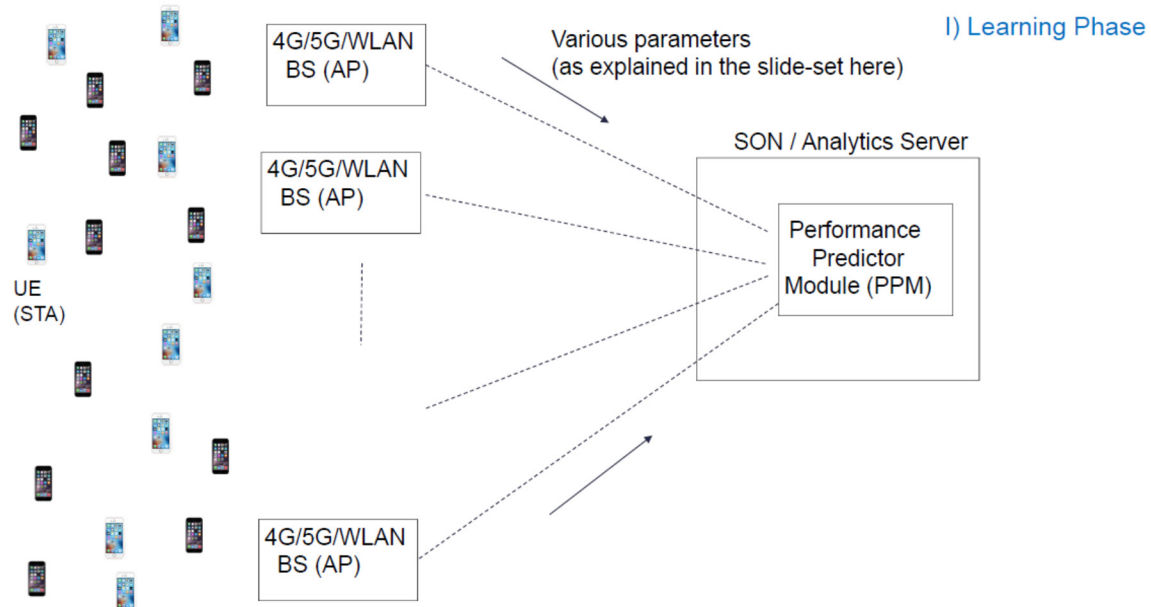


**FIG. 1**

In general, the ML methods presented herein enable the PPM-SON module to predict, for example, a degraded QoS class (QCI in LTE, Access Class in WLAN) that can be offered to a UE, if its desired QoS class (or QCI or AC) cannot be offered. The PPM-

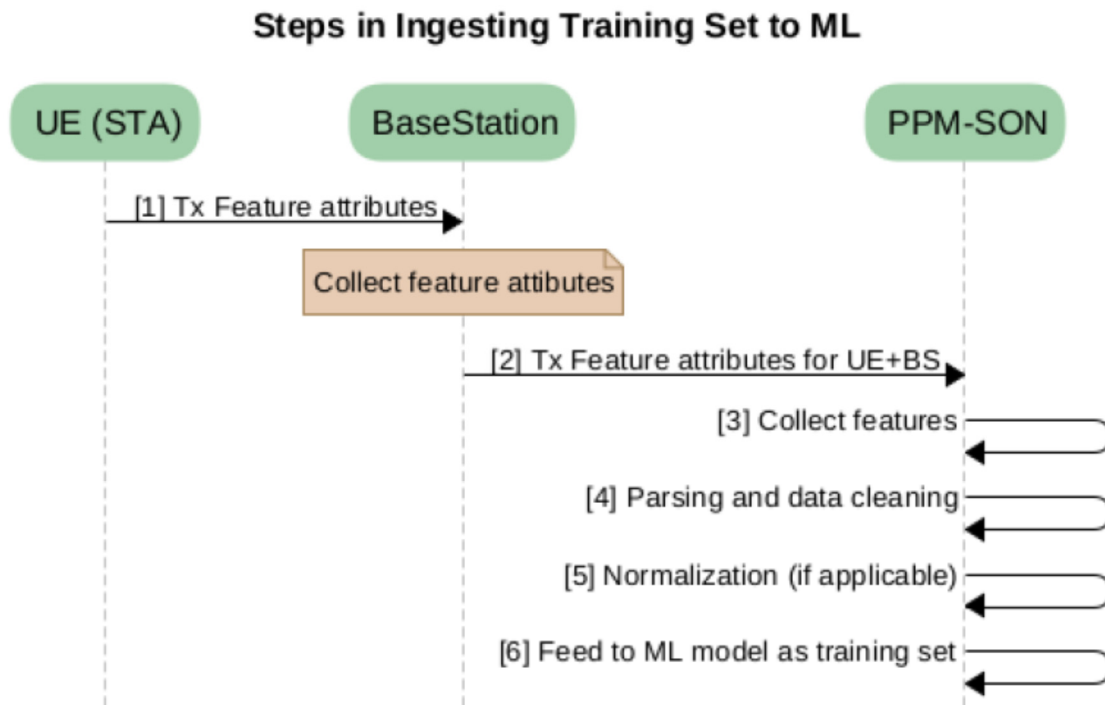
SON module can also learn to predict the class or QoS parameters, as needed. An option may also be provided where the PPM-SON module informs the UE and its originating base station whether or not neighboring base stations can meet the desired QoS class (or parameters). This information can be used by the UE or the originating base station to take policy based decisions.

FIG. 2, below, generally illustrates a learning phase in accordance with the techniques presented herein.



**FIG. 2**

FIG. 3, below, generally illustrates the steps in ingesting a training set in ML, in accordance with the techniques presented herein.

**FIG. 3**

During the feedback phase, the PPM-SON determines whether or not QoS requirements (e.g., QoS parameters of corresponding class) of an application can be met. As noted, in accordance with the techniques presented herein, the corresponding base station (or AP) provides parameters as in feature list to the PPM-SON which uses its models to predict type of QoS that can be provided to this UE/STA/applications. Feedback may also be provided as to whether or not neighboring base stations can meet its QoS requirements.

FIGs. 4 and 5, below, each generally illustrate different feedback phases, in accordance with the techniques presented herein.



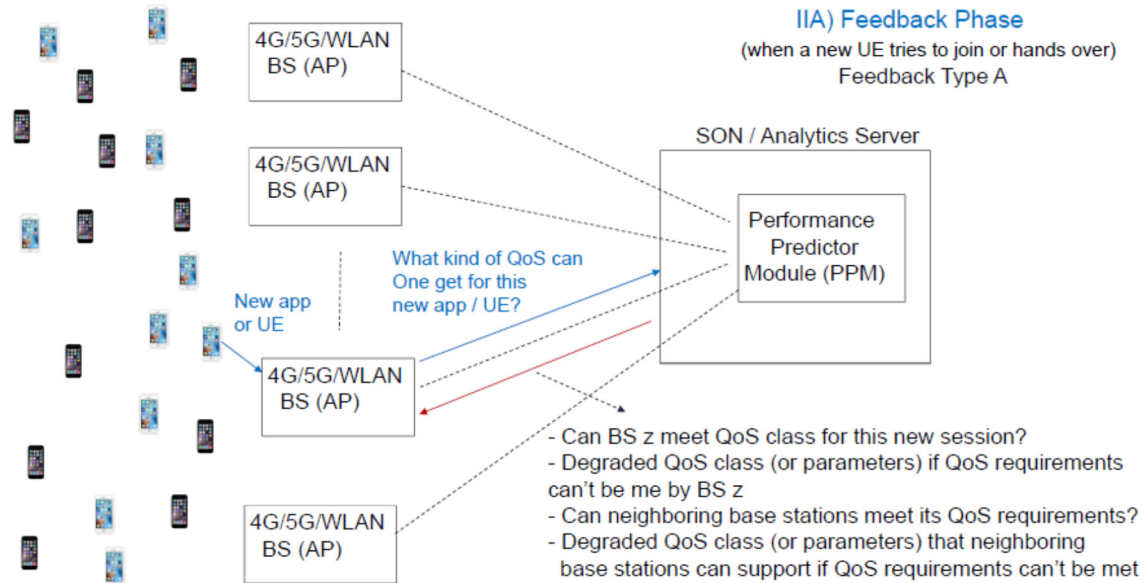


FIG. 4

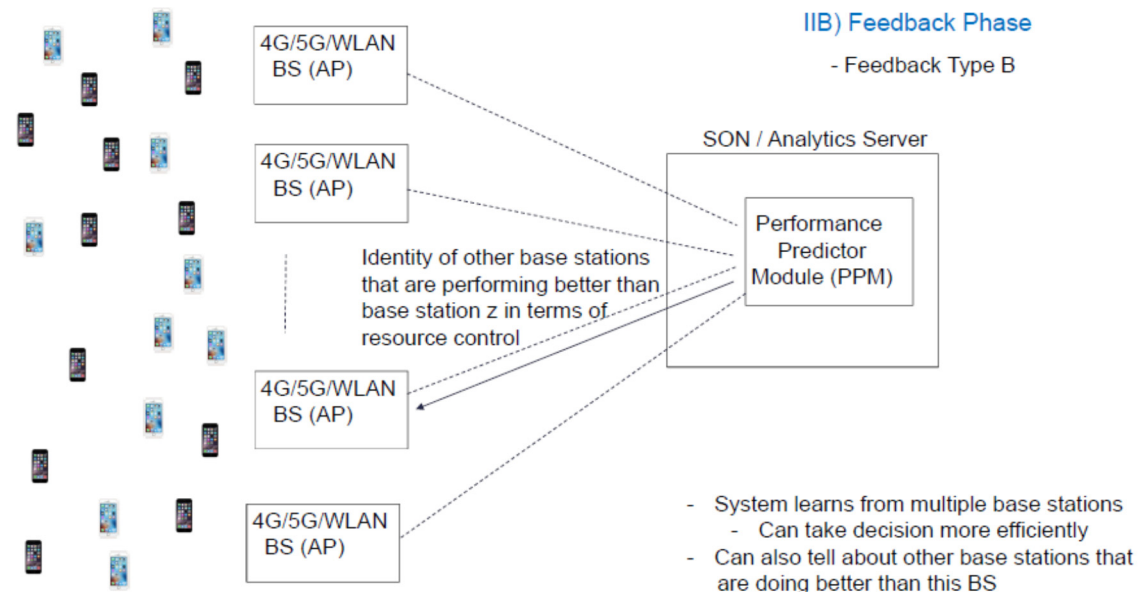
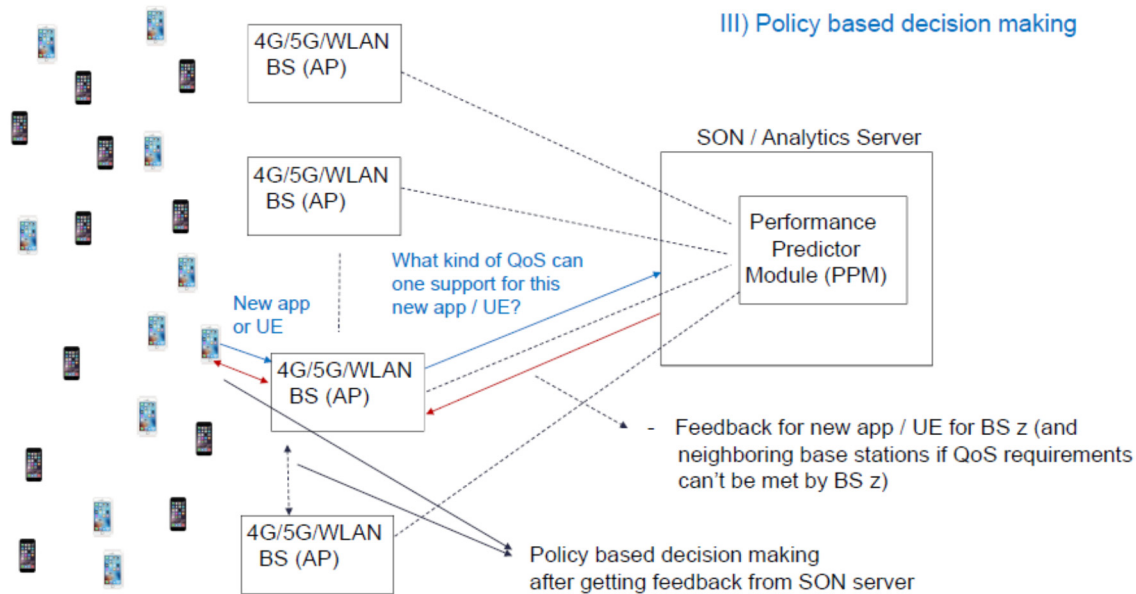


FIG. 5

The techniques presented herein enable policy based decisions. For example, assuming a UE referred to as "UE<sub>x</sub>" starts a new application session (as in 4G or starts communicating data as in WLAN) via a base station referred to as "BS<sub>z</sub>." BS<sub>z</sub>

communicates with the PPM-SON and gets feedback as above. If this feedback indicates that there are enough resources available at BS<sub>z</sub> to provide required performance of this application, it continues as usual. If the feedback indicates other possibilities (e.g., degraded QoS class for this application via BS<sub>z</sub> or possible QoS class via neighboring base stations) as in the feedback phase above, certain policy based decisions can be taken by BS<sub>z</sub>, UEx, and neighboring base stations. FIG. 6, below, generally illustrates policy based decision making, in accordance with the techniques presented herein.



**FIG. 6**

In the techniques presented herein, each BS continues collecting performance data of different applications and can contact PPM-SON module with “what-if” questions/scenarios. In these examples, the PPM-SON may also provide feedback for these various what-if scenarios.

FIG. 7, below, generally illustrates an arrangement for feedback on what-if scenarios, in accordance with the techniques presented herein.

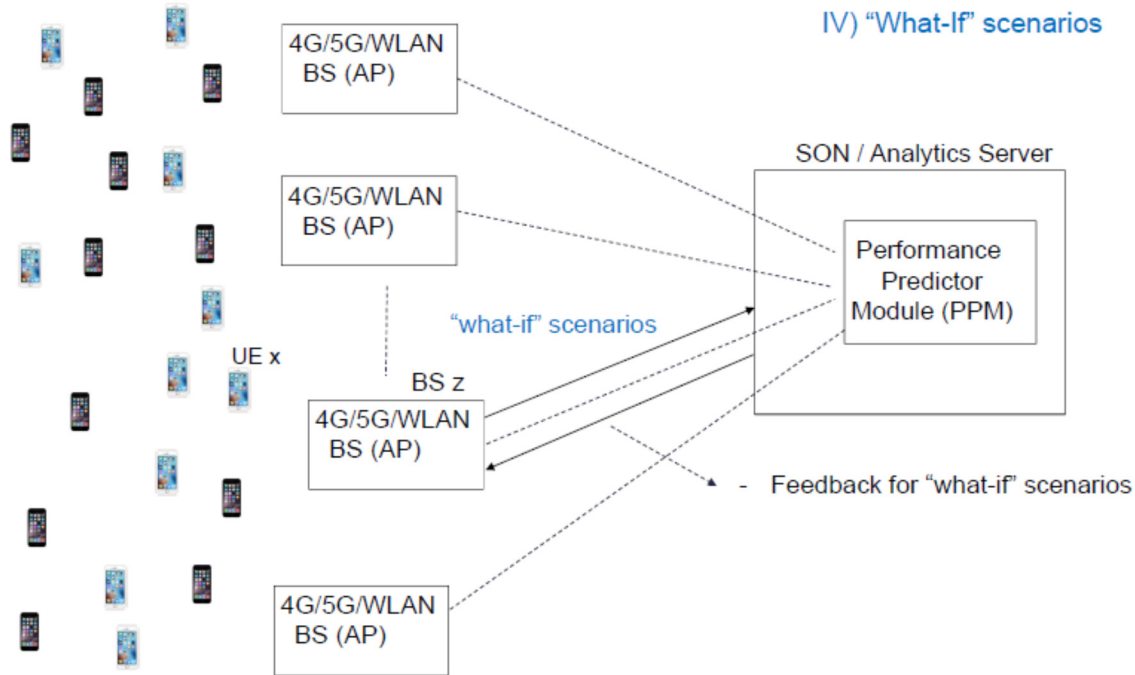


FIG. 7

For example, if a BS finds that  $p1\%$  of its delay sensitive applications are not meeting their delay / jitter requirements and  $p1$  is above a pre-specified threshold ( $p1\_thresh$ ) for that the BS, it contacts the PPM-SON to find a solution. In one method, the PPM-SON module finds a set of non-delay-sensitive applications for which QoS class can be changed (or degraded) to ensure that  $p1 < p1\_thresh$  and the PPM-SON informs the BS of this set.

In one example, there are four (4) QoS classes supported by a BS, namely: class 1 ( $c1$ ) which is a delay sensitive class, as well as classes 2 ( $c2$ ), 3 ( $c3$ ), and 4 ( $c4$ ), which are each non-delay-sensitive classes. The number of applications for each QoS class is:  $n_k$  (where  $k = 1, 2, 3$  and  $4$ ). The number of delay sensitive applications corresponding to class  $c1 = n1$ , and the number of non-delay sensitive applications for that BS =  $n2+n3+n4$ . In this case,  $p1 > p1\_thresh$  and this BS asks the PPM-SON for ways to make  $p1 < p1\_thresh$ .

In this example, the PPM-SON considers various options. In one such option, PPM-SON uses the previously learned models and provides updated values of number of applications for other QoS classes (e.g.,  $n2\_updated$ ,  $n3\_updated$  and  $n4\_updated$ ) to the BS so that the percentage of delay sensitive applications for which performance requirements are not met becomes less than a pre-specified threshold (i.e.  $p1 < p1\_thresh$

holds good again). The PPM-SON also provides identities of the applications for which QoS class can be changed.

In the above case, the BS could also provide different values of number of applications for certain QoS classes (such as  $n2\_new$ ,  $n4\_new$ ,  $n2$  is unchanged) and ask PPM-SON module if this would help to bring fraction of delay sensitive applications for which performance requirements are not met below a pre-specified threshold (i.e.  $p1 < p1\_thresh$ ). In this case, PPM-SON module will use the feature set with updated values of  $n2$  and  $n4$  (i.e.  $n2\_new$  and  $n4\_new$ ) and analyze this using its models to see if it results in  $p1 < p1\_thresh$  (and indicate this to BS).

In accordance with the techniques presented herein, BSz could ask the PPM-SON to analyze performance of other base stations (e.g., BSx and BSy) if it were to request  $u1$  UEs to move to BSx and  $u2$  UEs to move to BSy. The PPM-SON module will use feature set for each of these base stations with updated parameters, analyze performance using the learned models and provide feedback.

In summary, presented herein are techniques determine whether or not a base station can meet performance requirements of a newly added (or handed over) application / UE in a 4G/5G/WLAN system. The techniques presented herein collect variety of parameters from different base stations, analyze these at a Performance Predictor Module (PPM-SON) and learn how the resource management methods of a base station are behaving. The PPM-SON performs data cleaning to extract useful attributes, parsing to binary encode the attributes and uses ML models like SVM/SVR for training with the feature set and target variables specified, as noted above. As a new UE (or a new application of an existing UE or for a UE that is handing over from a different base station) joins a particular base station, that base station contacts the PPM-SON with relevant performance attributes for that application. The PPM-SON uses the previously learned models (i.e., learned in the learning phase) and informs the results to that base station (which the base station can eventually inform to that UE), hence providing a closed loop. The PPM-SON can, for example, inform the base station: (1) whether it can meet QoS requirements (QoS class and/or parameters) of that app/UE; (2) a degraded QoS class if it can't meet the asked QoS class, (3) what will be impact on performance of other applications (or class of applications) if this new application is admitted, (4) whether neighboring base stations can meet QoS

requirements (and degraded QoS class if they also can't meet QoS requirements of that app/UE). The base station, in turn, uses this information to either accept a new session or take decisions based on the policies provided at that BS and UE. It is noted that a typical resource control method works locally in a base station (or at a cloud server as part of cloud-RAN). Here, PPM-SON learns from multiple base stations, accelerates the decision making process and provides answer to questions related to some "what-if" scenarios (as explained above). PPM-SON also indicates to the originating base station about the identity (such as vendor id if allowed by the operator, software release id, location coordinates) of other base stations which are performing better than this base station in terms of resource control. SON protocols (between BS and SON server) and L2 (or higher layer) protocols between UE and BS may be updated to carry various parameters.